

Curve interpolation and shape modeling via probabilistic nodes combination

Dariusz Jacek Jakóbczak

Received: 30 September 2013 / Accepted: 8 February 2014 / Published online: 23 April 2014
© The Author(s) 2014. This article is published with open access at Springerlink.com

Abstract The proposed method, called probabilistic nodes combination (PNC), is the method of 2D curve interpolation and modeling using the set of key points (knots or nodes). Nodes can be treated as characteristic points of the object for modeling. The model of each individual symbol or data can be built by choice of probability distribution function and nodes combination. PNC modeling via nodes combination and parameter γ as probability distribution function enables curve parameterization and interpolation for each specific data or handwritten symbol. Two-dimensional curve is modeled and interpolated via nodes combination and different functions as discrete or continuous probability distribution functions: polynomial, sine, cosine, tangent, cotangent, logarithm, exponent, arcsin, arccos, arctan, arccot or power function. The novelty of the paper consists of two generalizations: generalization of previous MHR method with various nodes combinations and generalization of linear interpolation with different (no basic) probability distribution functions and nodes combinations.

Keywords Curve interpolation · PNC method · Shape representation · Contour parameterization · Nodes combination · Probabilistic modeling

1 Introduction

Probabilistic modeling is still a developing branch of computer science: operational research (for example probabilistic model-based prognosis) [1], decision making techniques and

probabilistic modeling [2], artificial intelligence and machine learning. Different aspects of probabilistic methods are used: stochastic processes and stochastic model-based techniques, Markov processes [3], Poisson processes, Gamma processes, a Monte Carlo method, Bayes rule, conditional probability and many probability distributions. In this paper the goal of probability distribution function is to describe the position of unknown points between the given interpolation nodes. two-dimensional curve (opened or closed) is used to represent the data points.

The paper clarifies the significance and novelty of the proposed method compared to existing methods (for example, polynomial interpolations and Bézier curves in Sect. 2.1). Previous published papers of the author dealt with the method of Hurwitz–Radon matrices (MHR method). The novelty of this paper and the proposed method consists in the fact that calculations are free from the family of Hurwitz–Radon matrices. The problem statement of this paper is: *how to reconstruct (interpolate) missing points of 2D curve having a set of interpolation nodes (key points) and using the information about probabilistic distribution of unknown points*. For example, the simplest basic distribution leads to the easiest interpolation—linear interpolation. Apart from probability distribution, additionally there is the second factor of the proposed interpolation method: nodes combination. The simplest nodes combination is zero. Thus, the proposed curve modeling is based on two agents: probability distribution and nodes combination. The first trial of probabilistic modeling in the MHR version was described in [4]. The significance of this paper consists in generalization for the MHR method: the computations are done without matrices in curve fitting and shape modeling, with clear point interpolation formula based on probability distribution function (continuous or discrete) and nodes combination. The paper also consists of generalization for linear interpolation with different (no basic)

D. J. Jakóbczak (✉)
Department of Electronics and Computer Science,
Technical University of Koszalin, Sniadeckich 2,
75-453 Koszalin, Poland
e-mail: djakob@ie.tu.koszalin.pl; dariusz.jakobczak@tu.koszalin.pl

probability distribution functions and nodes combinations. So this paper answers the question: “Why and when should we use PNC method?”.

Curve interpolation [5] represents one of the most important problems in mathematics and computer science: how do we model the curve [6] via a discrete set of two-dimensional points [7]? Also the matter of shape representation (as closed curve—contour) and curve parameterization is still open [8]. For example, pattern recognition, signature verification or handwriting identification problems are based on curve modeling via the choice of key points. So interpolation is not only a pure mathematical problem, but important task in computer vision and artificial intelligence. The paper wants to approach a problem of curve modeling by characteristic points. The proposed method relies on nodes combination and functional modeling of curve points situated between the basic set of key points. The functions that are used in calculations represent a whole family of elementary functions with inverse functions: polynomials, trigonometric, cyclometric, logarithmic, exponential and power function. These functions are treated as probability distribution functions in the range [0; 1].

2 Shape representation and curve reconstruction

An important problem in machine vision and computer vision [9] is that of appropriate shape representation and reconstruction. Classical discussion about shape representation is based on the problem: contour versus skeleton. This paper votes for contour which forms the boundary of the object. The contour of the object, represented by contour points, consists of information which allows us to describe many important features of the object as shape coefficients [10]. In the paper, contour deals with a set of curves. Curve modeling and generation is a basic subject in many branches of industry and computer science, for example in the CAD/CAM software.

The representation of shape has a great impact on the accuracy and effectiveness of object recognition [11]. In the literature, shape has been represented by many options including curves [12], graph-based algorithms and medial axis [13] to enable shape-based object recognition. Digital curve (open or closed) can be represented by chain code (Freeman’s code). Chain code depends on selection of the started point and transformations of the object. So Freeman’s code is one of the methods to describe and find the contour of the object. An analog (continuous) version of Freeman’s code is the curve $\alpha - s$. Another contour representation and reconstruction is based on Fourier coefficients calculated in discrete Fourier transformation (DFT). These coefficients are used to fix the similarity of the contours with different sizes or directions. If we assume that the contour is built from segments of a line and fragments of circles or ellipses, Hough transforma-

tion is applied to detect the contour lines. Also, geometrical moments of the object are used during the process of object shape representation [14].

2.1 A comparative analysis with other interpolation methods (why only polynomials?)

All interpolation theory is based on polynomials. But why? Many kinds of polynomials are used for interpolation: classical polynomials, trigonometric polynomials, orthogonal polynomials (Tschebyscheff, Legendre, Laguerre), rational polynomials. But what about the exceptional situations with unexpected features of curve, data or nodes. Then polynomials are not the solution, for example when:

1. The curve is not a graph of function (no matter—open or closed curve).
2. The curve does not have to be smooth at the interpolation nodes: for example, curve representing symbols, signature, handwriting or other specific data.
3. Nodes are fixed and there is no possibility of choosing “better” nodes as for orthogonal polynomials.
4. The curve differs considerably from any interpolation polynomial.
5. The curve fails to be differentiable at some points.
6. between each pair of nodes we are not interested in linear interpolation (basic probability distribution and zero nodes combination), but there ought to be some generalization (even for two nodes only) with other probability distributions and nodes combinations.
7. Interpolated points depend on some chosen nodes (two nearest nodes or more) via nodes combination $h(p_1, p_2, \dots, p_m)$ in (1).
8. We are not interested in the formula of interpolation function (for lower computational costs), but only calculated points of modeled curve are ready to be used in numerical computations.
9. The formula of curve or function is known, but for some reason (for example, high computational costs or hard polynomial interpolation), the curve has to be modeled or fitted in some way for numerical calculations—the examples for PNC interpolation (in MHR version) of functions $f(x) = 2/x$ and $f(x) = 1/(1+5x^2)$ with quantified measures and experimental comparison with classical polynomial interpolation in [15].
10. Extrapolation problem is also a big numerical challenge and PNC interpolation enables the extension into extrapolation [16] with α outside of [0; 1] and $\gamma = F(\alpha)$ still strictly monotonic, $F(0) = 0$, $F(1) = 1$. So for example $\gamma = \alpha^2$ is impossible for extrapolation if $\alpha \leq 0$ [17]. Polynomial or other interpolations are sometimes useless for extrapolation.

11. Having only nodes the user may have “negative” information (from specific character of data): no polynomial interpolation.
12. All calculations are numerical (discrete)—even $\gamma = F(\alpha)$ is to be given in tabular (discrete) form. There is no need to build continuous function: polynomial or others.
13. The parametric version of the modeled curve is to be found.

The above 13 important and heavy individual and characteristic features of some curves and their interpolations show that there may exist situations with unexpected assumptions for interpolation.

Why not classical interpolation? Classical methods are useless to interpolate the function that fails to be differentiable at one point, for example the absolute value function $f(x) = |x|$ at $x = 0$. If point $(0; 0)$ is one of the interpolation nodes, then precise polynomial interpolation of the absolute value function is impossible. Also, when the graph of the interpolated function differs from the shape of the polynomial considerably, for example $f(x) = 1/x$, interpolation is very hard because of existing local extrema and the roots of the polynomial. We cannot forget about the Runge’s phenomenon: when nodes are equidistance then high-order polynomial oscillates toward the end of the interval, for example close to -1 and 1 with function $f(x) = 1/(1 + 25x^2)$ [7]. These classical negative cases do not appear in the proposed PNC method. Experimental comparison for PNC with polynomial interpolation is to be found in [15, 18].

Nowadays, methods apply mainly polynomial functions in different versions (trigonometric, orthogonal, rational) and for example Bernstein polynomials in Bézier curves, splines [19] and NURBS [20]. But Bézier curves do not represent the interpolation method (rather interpolation-approximation method) and cannot be used for example in handwriting modeling with key points (interpolation nodes). In comparison, the PNC method with Bézier curves, Hermite curves and B-curves (*B-splines*) or NURBS has one unpleasant feature: small change of one characteristic point can result in unwanted change of the whole reconstructed curve. Such a feature does not appear in the proposed PNC method which is more stable than Bézier curves. Only the first and last characteristic points are situated on the Bézier curve (interpolation), the rest of the characteristic points lay outside the Bézier curve (approximation). Numerical methods for data interpolation are based on polynomial or trigonometric functions, for example Lagrange, Newton, Aitken and Hermite methods. These methods have many weak sides [21] and are not sufficient for curve interpolation in the situations when the curve cannot be built by polynomials or trigonometric functions. Also, there exist several well-established methods of curve modeling, for example shape-preserving techniques

[22], subdivision algorithms [23] and others [24] to overcome the difficulties of polynomial interpolation, but probabilistic interpolation with nodes combination seems to be quite novel in the area of shape modeling. The proposed 2D curve interpolation is the functional modeling via any elementary functions and it helps us to fit the curve during the computations.

This paper presents novel probabilistic nodes combination (PNC) method of curve interpolation. This paper takes up the new PNC method of two-dimensional curve modeling via the examples using the family of Hurwitz–Radon matrices (MHR method) [25], but not only that (other nodes combinations). The method of PNC requires minimal assumptions: the only information about a curve is the set of at least two nodes. The proposed PNC method is applied to curve modeling via different coefficients: polynomial, sinusoidal, cosinusoidal, tangent, cotangent, logarithmic, exponential, arcsin, arccos, arctan, arccot or power. The function for PNC calculations is chosen individually at each interpolation and represents the probability distribution function of parameter $\alpha \in [0; 1]$ for every point situated between two interpolation knots. The PNC method uses two-dimensional vectors (x, y) for curve modeling—knots $p_i = (x_i, y_i) \in \mathbf{R}^2$ in the PNC method, $i = 1, 2, \dots, n$:

1. PNC needs two knots or more ($n \geq 2$).
2. If the first node and the last node are the same ($p_1 = p_n$), then the curve is closed (contour).
3. For more precise modeling, knots ought to be settled at key points of the curve, for example local minimum or maximum and at least one node between two successive local extrema.

Condition 3 means for example the highest point of the curve in a particular orientation, convexity changing or curvature extrema. So this paper wants to answer the question: how do we interpolate the curve by a set of knots [26]?

Nodes on Fig. 1 represent the characteristic points of the handwritten letter or symbol: if $n = 5$ then the curve is open and if $n = 6$ then the curve is closed (contour). The examples of PNC curve modeling for these nodes are described

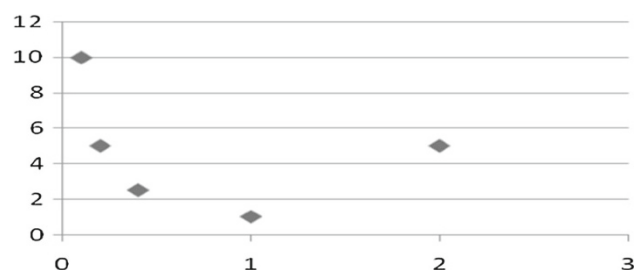


Fig. 1 Five knots of the curve before modeling

later in this paper (Sect. 3). The coefficients for PNC curve modeling are computed using nodes combinations and probability distribution functions: polynomials, power functions, sine, cosine, tangent, cotangent, logarithm, exponent or arcsin, arccos, arctan or arccot.

3 Novelty of probabilistic interpolation and modeling technique

The method of PNC enables computing points between two successive nodes of the curve: calculated points are interpolated and parameterized for real number $\alpha \in [0; 1]$ in the range of two successive nodes. The PNC method uses the combinations of nodes $p_1 = (x_1, y_1)$, $p_2 = (x_2, y_2)$, \dots , $p_n = (x_n, y_n)$ as $h(p_1, p_2, \dots, p_m)$ and $m = 1, 2, \dots, n$. Nodes combination h is defined individually for each curve to interpolate points (x, y) with second coordinate $y = y(c)$ for any first coordinate $x = c$ situated between nodes (x_i, y_i) and (x_{i+1}, y_{i+1}) :

$$c = \alpha \cdot x_i + (1 - \alpha) \cdot x_{i+1}, \quad i = 1, 2, \dots, n - 1,$$

$$y(c) = \gamma \cdot y_i + (1 - \gamma)y_{i+1} + \gamma(1 - \gamma) \cdot h(p_1, p_2, \dots, p_m),$$

$$\alpha \in [0; 1], \quad \gamma = F(\alpha) \in [0; 1]. \quad (1)$$

So, c and α represent the same—coordinate x of any point (x, y) between two successive nodes (x_i, y_i) and (x_{i+1}, y_{i+1}) : having c we can calculate α and vice versa. PNC curve modeling relies on two factors: function $\gamma = F(\alpha)$ and nodes combination $h(p_1, p_2, \dots, p_m)$. Function F is a probabilistic distribution function for random variable $\alpha \in [0; 1]$ and parameter γ leads PNC interpolation into probabilistic modeling. The second factor, the combination of nodes h , is responsible for making dependent a reconstructed point on the coordinates of several nodes. The simplest case is for $h = 0$. Here are the examples of h computed for the MHR method [18]:

$$h(p_1, p_2) = \frac{y_1}{x_1}x_2 + \frac{y_2}{x_2}x_1$$

(only two neighboring nodes are taken for PNC calculations) or

$$h(p_1, p_2, p_3, p_4) = \frac{1}{x_1^2 + x_3^2}(x_1x_2y_1 + x_2x_3y_3 + x_3x_4y_1 - x_1x_4y_3) + \frac{1}{x_2^2 + x_4^2}(x_1x_2y_2 + x_1x_4y_4 + x_3x_4y_2 - x_2x_3y_4)$$

(more than two neighboring nodes are used in PNC interpolation).

The examples of other nodes combinations are presented in Sect. 3. Formula (1) represents curve parameterization $(x(\alpha), y(\alpha))$ between two successive nodes (x_i, y_i) and

(x_{i+1}, y_{i+1}) as $\alpha \in [0; 1]$:

$$x(\alpha) = \alpha \cdot x_i + (1 - \alpha) \cdot x_{i+1}$$

and

$$y(\alpha) = F(\alpha) \cdot y_i + (1 - F(\alpha))y_{i+1} + F(\alpha)(1 - F(\alpha)) \cdot h(p_1, p_2, \dots, p_m),$$

$$y(\alpha) = F(\alpha) \cdot (y_i - y_{i+1} + (1 - F(\alpha)) \cdot h(p_1, p_2, \dots, p_m)) + y_{i+1}.$$

The proposed parameterization gives us an infinite number of possibilities for curve calculations (determined by choice of F and h) as there is an infinite number of human handwritten letters and symbols. Nodes combination is the individual feature of each modeled curve (for example a handwritten character). Coefficient $\gamma = F(\alpha)$ and nodes combination h are key factors in PNC curve interpolation and shape modeling.

3.1 Distribution functions in PNC interpolation and curve fitting

Points settled between the nodes are computed using the PNC method. Each real number $c \in [a; b]$ is calculated by a convex combination $c = \alpha \cdot a + (1 - \alpha) \cdot b$ for

$$\alpha = \frac{b - c}{b - a} \in [0; 1].$$

The key question is dealing with coefficient γ in (1). The simplest way of PNC calculation means $h = 0$ and $\gamma = \alpha$ (basic probability distribution). Then, PNC represents a linear interpolation. The MHR method [27] is not a linear interpolation. MHR [15] is an example of PNC modeling.

Each interpolation requires specific distribution of parameter α and γ (1) depending on parameter $\alpha \in [0; 1]$:

$$\gamma = F(\alpha), \quad F : [0; 1] \rightarrow [0; 1], \quad F(0) = 0, \quad F(1) = 1$$

and F is strictly monotonic.

Coefficient γ is calculated using different functions (polynomials, power functions, sine, cosine, tangent, cotangent, logarithm, exponent, arcsin, arccos, arctan or arccot, also inverse functions) and the choice of function is connected with initial requirements and curve specifications. Different values of coefficient γ are connected with applied functions $F(\alpha)$. The functions (2)–(34) represent the examples of probability distribution functions for random variable $\alpha \in [0; 1]$ and real number $s > 0$:

1. power function

$$\gamma = \alpha^s \quad \text{with } s > 0. \quad (2)$$

For $s = 1$: basic version of PNC and MHR [28] methods when $\gamma = \alpha$.

2. sine

$$\gamma = \sin(\alpha^s \cdot \pi/2), \quad s > 0 \quad (3)$$

or

$$\gamma = \sin^s(\alpha \cdot \pi/2), \quad s > 0. \quad (4)$$

$$\text{For } s = 1 : \gamma = \sin(\alpha \cdot \pi/2). \quad (5)$$

3. cosine

$$\gamma = 1 - \cos(\alpha^s \cdot \pi/2), \quad s > 0 \quad (6)$$

or

$$\gamma = 1 - \cos^s(\alpha \cdot \pi/2), \quad s > 0. \quad (7)$$

$$\text{For } s = 1 : \gamma = 1 - \cos(\alpha \cdot \pi/2). \quad (8)$$

4. tangent

$$\gamma = \tan(\alpha^s \cdot \pi/4), \quad s > 0 \quad (9)$$

or

$$\gamma = \tan^s(\alpha \cdot \pi/4), \quad s > 0. \quad (10)$$

$$\text{For } s = 1 : \gamma = \tan(\alpha \cdot \pi/4). \quad (11)$$

5. logarithm

$$\gamma = \log_2(\alpha^s + 1), \quad s > 0 \quad (12)$$

or

$$\gamma = \log_2^s(\alpha + 1), \quad s > 0. \quad (13)$$

$$\text{For } s = 1 : \gamma = \log_2(\alpha + 1). \quad (14)$$

6. exponent

$$\gamma = \left(\frac{a^\alpha - 1}{a - 1} \right)^s, \quad s > 0 \text{ and } a > 0 \text{ and } a \neq 1. \quad (15)$$

$$\text{For } s = 1 \text{ and } a = 2 : \gamma = 2^\alpha - 1. \quad (16)$$

7. arcsine

$$\gamma = 2/\pi \cdot \arcsin(\alpha^s), \quad s > 0 \quad (17)$$

or

$$\gamma = (2/\pi \cdot \arcsin \alpha)^s, \quad s > 0. \quad (18)$$

$$\text{For } s = 1 : \gamma = 2/\pi \cdot \arcsin(\alpha). \quad (19)$$

8. arccosine

$$\gamma = 1 - 2/\pi \cdot \arccos(\alpha^s), \quad s > 0 \quad (20)$$

or

$$\gamma = 1 - (2/\pi \cdot \arccos \alpha)^s, \quad s > 0. \quad (21)$$

$$\text{For } s = 1 : \gamma = 1 - 2/\pi \cdot \arccos(\alpha). \quad (22)$$

9. arctangent

$$\gamma = 4/\pi \cdot \arctan(\alpha^s), \quad s > 0 \quad (23)$$

or

$$\gamma = (4/\pi \cdot \arctan \alpha)^s, \quad s > 0. \quad (24)$$

$$\text{For } s = 1 : \gamma = 4/\pi \cdot \arctan(\alpha). \quad (25)$$

10. cotangent

$$\gamma = \text{ctg}(\pi/2 - \alpha^s \cdot \pi/4), \quad s > 0 \quad (26)$$

or

$$\gamma = \text{ctg}^s(\pi/2 - \alpha \cdot \pi/4), \quad s > 0. \quad (27)$$

$$\text{For } s = 1 : \gamma = \text{ctg}(\pi/2 - \alpha \cdot \pi/4). \quad (28)$$

11. arccotangent

$$\gamma = 2 - 4/\pi \cdot \text{arcctg}(\alpha^s), \quad s > 0 \quad (29)$$

or

$$\gamma = (2 - 4/\pi \cdot \text{arcctg} \alpha)^s, \quad s > 0. \quad (30)$$

$$\text{For } s = 1 : \gamma = 2 - 4/\pi \cdot \text{arcctg}(\alpha). \quad (31)$$

Functions used in γ calculations (2)–(31) are strictly monotonic for random variable $\alpha \in [0; 1]$ as $\gamma = F(\alpha)$ is a probability distribution function. Also, inverse function $F^{-1}(\alpha)$ is appropriate for γ calculations. The choice of function and value s depends on curve specifications and individual requirements.

The proposed (2)–(31) probability distributions are continuous, but of course parameter γ can represent discrete probability distributions, for example: $F(0.1) = 0.23$, $F(0.2) = 0.3$, $F(0.3) = 0.42$, $F(0.4) = 0.52$, $F(0.5) = 0.63$, $F(0.6) = 0.69$, $F(0.7) = 0.83$, $F(0.8) = 0.942$, $F(0.9) = 0.991$. What is very important in the PNC method is that two curves (for example a handwritten letter) may have the same

set of nodes, but different h or γ results in different interpolations (Figs. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18).

The algorithm of PNC interpolation and modeling (1) is as follows:

Step 1: Choice of knots p_i at key points.

Step 2: Choice of nodes combination $h(p_1, p_2, \dots, p_m)$.

Step 3: Choice of distribution $\gamma = F(\alpha)$: (2)–(31) or others (continuous or discrete).

Step 4: Determining values of α : $\alpha = 0.1, 0.2, \dots, 0.9$ (nine points) or $0.01, 0.02, \dots, 0.99$ (99 points) or others.

Step 5: The computations (1).

These five steps can be treated as the algorithm of PNC method of curve modeling and interpolation (1). Without knowledge about the formula of curve or function, PNC inter-

polation has to implement the coefficients γ (2)–(31), but PNC is not limited only to these coefficients. Each strictly monotonic function F between points $(0; 0)$ and $(1; 1)$ can be used in PNC modeling.

4 Handwritten symbol modeling and curve fitting

Curve knots $p_1 = (0.1; 10)$, $p_2 = (0.2; 5)$, $p_3 = (0.4; 2.5)$, $p_4 = (1; 1)$ and $p_5 = (2; 5)$ from Fig. 1 are used in some examples of PNC method in handwritten character modeling. Figures 2, 3, 4, 5, 6, 7, 8, 9 represent PNC as MHR interpolation [29] with different γ . The points of the curve are calculated with no matrices ($N=1$) and $\gamma = \alpha$ in example 1 and with matrices of dimension $N=2$ in Examples 2–8 for $\alpha = 0.1, 0.2, \dots, 0.9$.

Fig. 2 PNC character modeling for nine reconstructed points between nodes

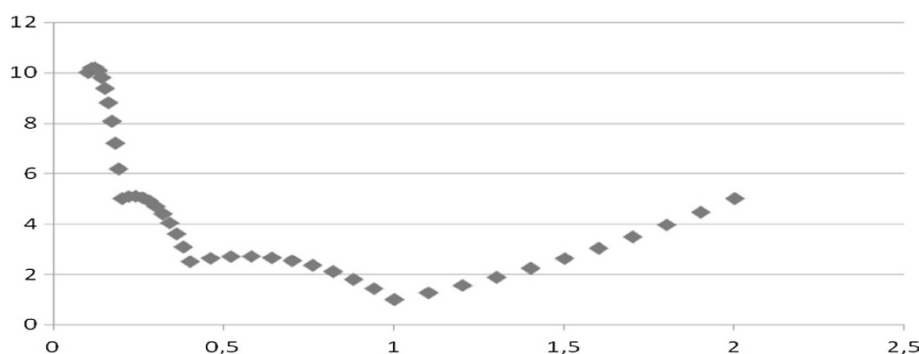


Fig. 3 Sinusoidal modeling with nine reconstructed curve points between nodes

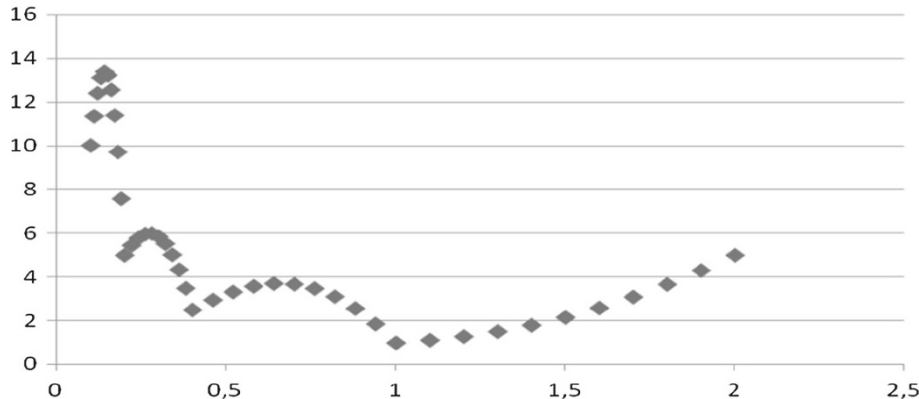


Fig. 4 Tangent character modeling with nine interpolated points between nodes

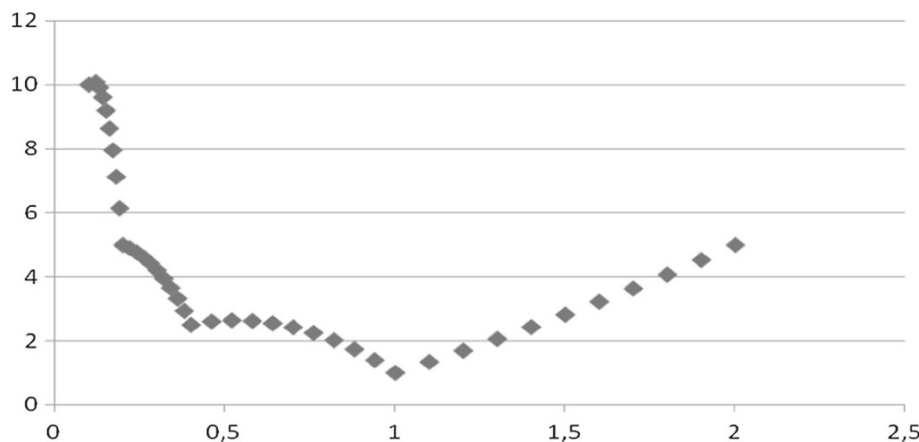


Fig. 5 Tangent curve modeling with nine recovered points between nodes

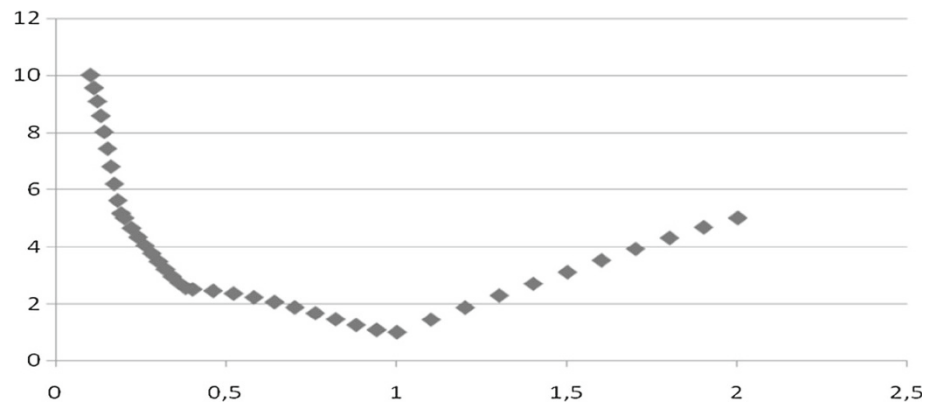


Fig. 6 Tangent symbol modeling with nine reconstructed points between nodes

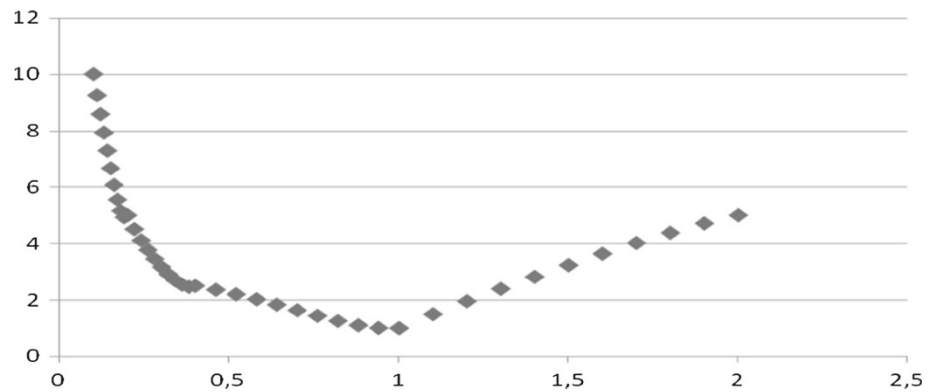
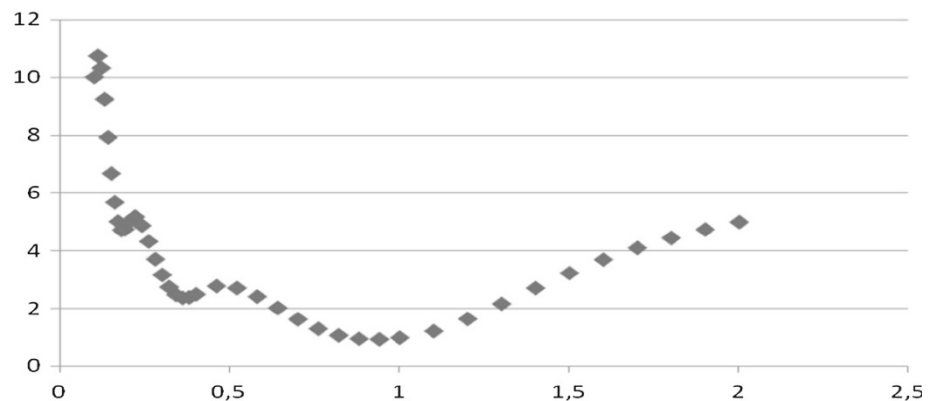


Fig. 7 Sinusoidal modeling with nine interpolated curve points between nodes



Example 1 PNC curve interpolation (1) for $\gamma = \alpha$ and

$$h(p_1, p_2) = \frac{y_1}{x_1}x_2 + \frac{y_2}{x_2}x_1 :$$

For $N = 2$ (Examples 2–8) MHR version [30] as PNC method gives us

$$\begin{aligned} h(p_1, p_2, p_3, p_4) &= \frac{1}{x_1^2 + x_3^2}(x_1x_2y_1 + x_2x_3y_3 + x_3x_4y_1 - x_1x_4y_3) \\ &+ \frac{1}{x_2^2 + x_4^2}(x_1x_2y_2 + x_1x_4y_4 + x_3x_4y_2 - x_2x_3y_4). \end{aligned}$$

Example 2 PNC sinusoidal interpolation with $\gamma = \sin(\alpha \cdot \pi/2)$.

Example 3 PNC tangent interpolation for $\gamma = \tan(\alpha \cdot \pi/4)$.

Example 4 PNC tangent interpolation with $\gamma = \tan(\alpha^s \cdot \pi/4)$ and $s = 1.5$.

Example 5 PNC tangent curve interpolation for $\gamma = \tan(\alpha^s \cdot \pi/4)$ and $s = 1.797$.

Example 6 PNC sinusoidal interpolation with $\gamma = \sin(\alpha^s \cdot \pi/2)$ and $s = 2.759$.

Example 7 PNC power function modeling for $\gamma = \alpha^s$ and $s = 2.1205$.

Example 8 PNC logarithmic curve modeling with $\gamma = \log_2(\alpha^s + 1)$ and $s = 2.533$.

Fig. 8 Power function curve modeling with nine recovered points between nodes

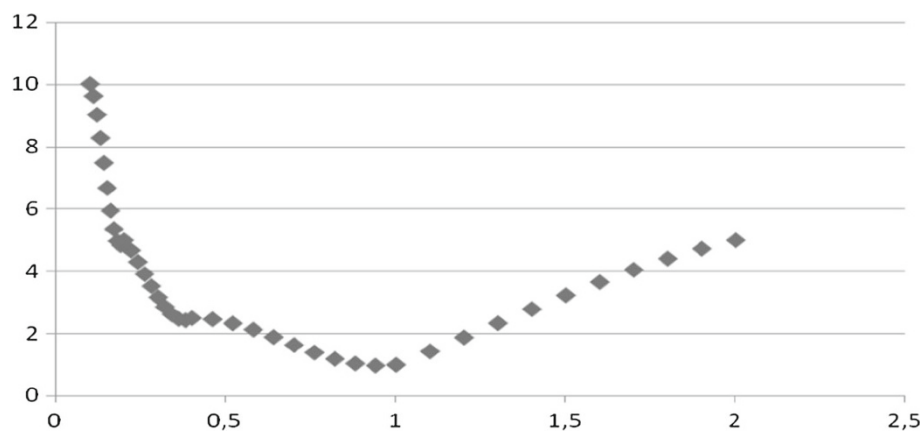


Fig. 9 Logarithmic character modeling with nine reconstructed points between nodes

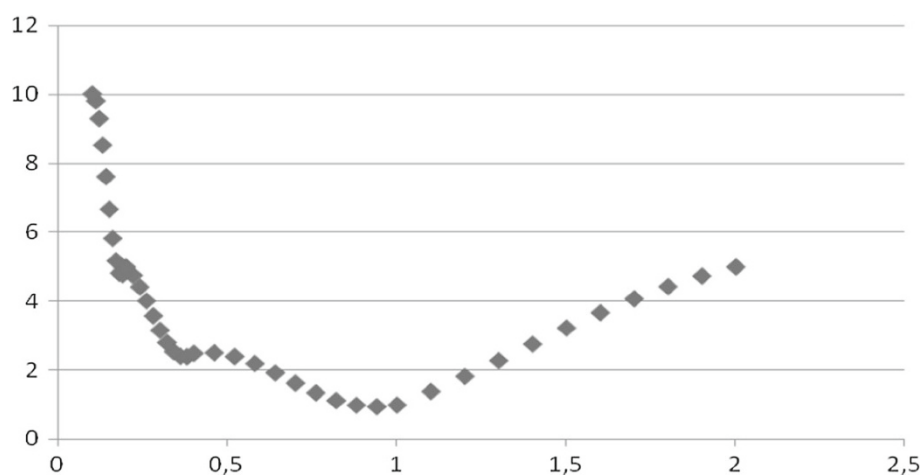


Fig. 10 Quadratic symbol modeling with nine reconstructed points between nodes

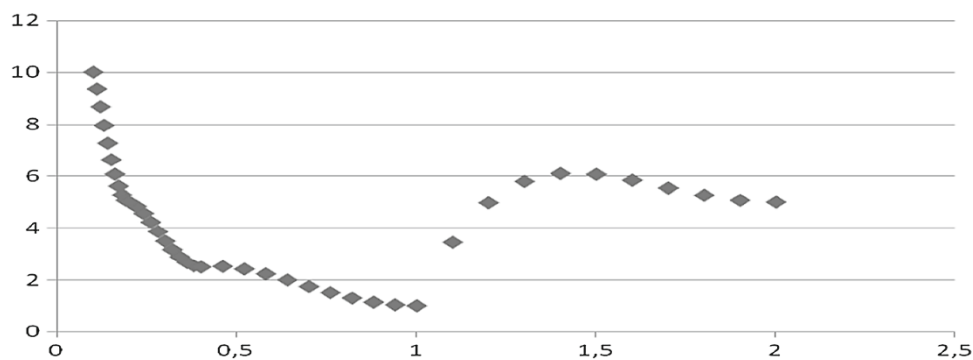


Fig. 11 Cubic character modeling with nine reconstructed points between nodes

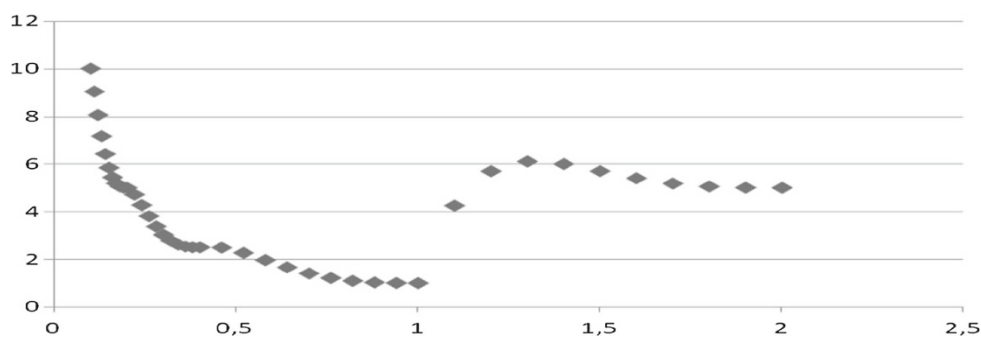


Fig. 12 Quadratic contour modeling with nine reconstructed points between nodes

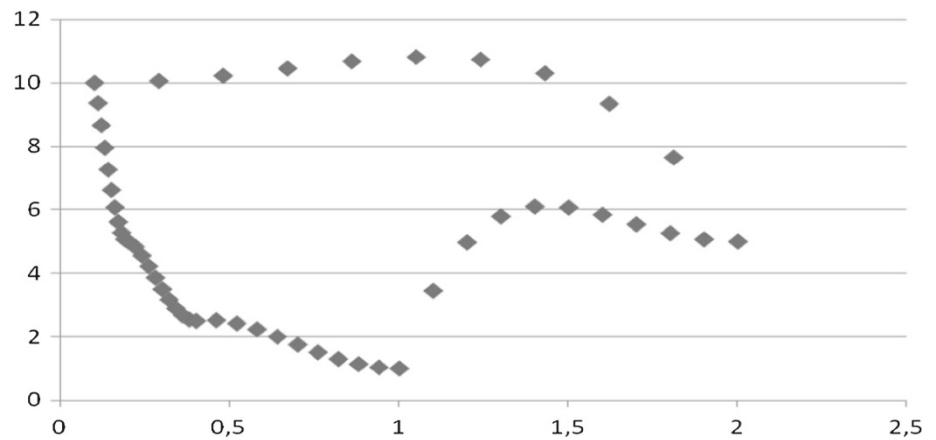


Fig. 13 Cubic shape modeling with nine reconstructed points between nodes

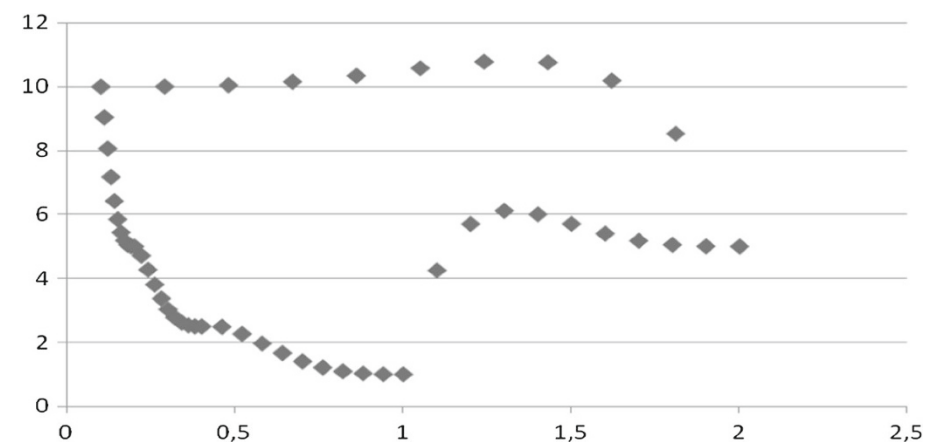
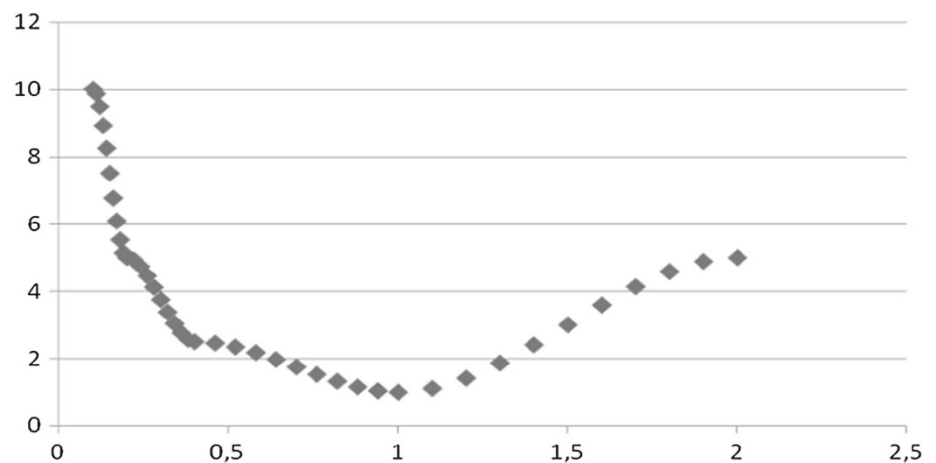


Fig. 14 Beta distribution in handwritten character modeling



These eight examples demonstrate the possibilities of PNC curve interpolation and handwritten character modeling for key nodes in the MHR version. Here are other examples of PNC modeling (but not MHR):

Example 9 PNC for $\gamma = \alpha^2$ and $h(p_1, p_2) = x_1 y_1 + x_2 y_2$:

Example 10 PNC for $\gamma = \alpha^3$ and $h(p_1, p_2) = x_1 y_1 + x_2 y_2$:

If we consider Fig. 1 as closed curve (contour) with the node $p_6 = p_1 = (0.1; 10)$, then Examples 9 and 10 give the shapes:

Example 11 PNC for $\gamma = \alpha^2$ and $h(p_1, p_2) = x_1 y_1 + x_2 y_2$:

Example 12 PNC for $\gamma = \alpha^3$ and $h(p_1, p_2) = x_1 y_1 + x_2 y_2$:

Every man has an individual style of handwriting. Recognition of handwritten letter or symbol needs modeling, and the model of each individual symbol or character can be built

Fig. 15 Beta distribution in handwritten symbol modeling

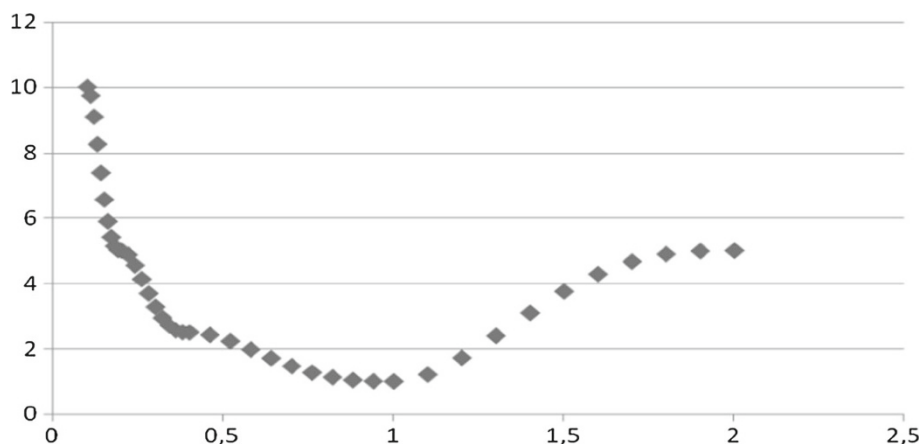


Fig. 16 Beta distribution in handwritten letter modeling

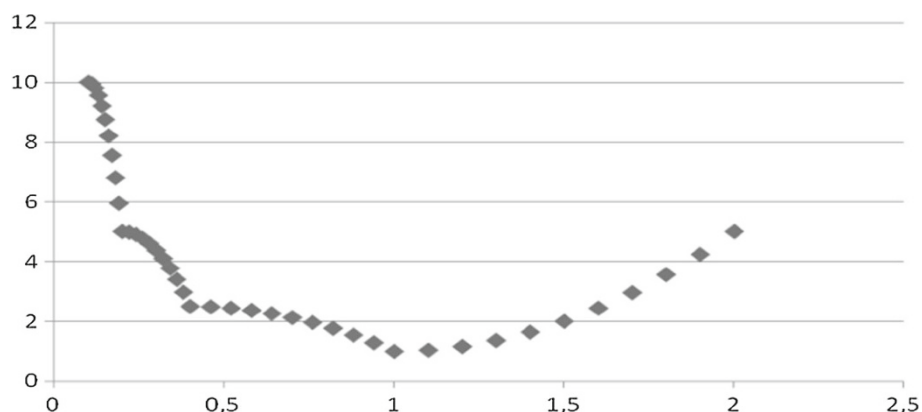
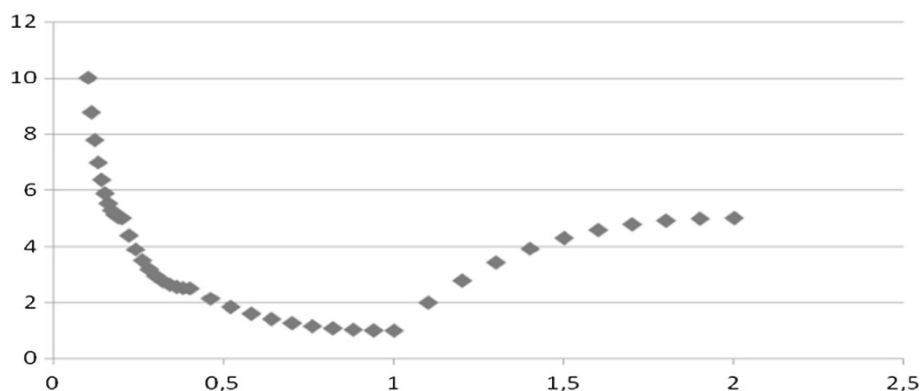


Fig. 17 Exponential distribution in handwritten character modeling



by choice of γ and h in (1). PNC modeling via nodes combinations h and parameter γ as probability distribution function enables curve interpolation for each specific letter or symbol.

The number of reconstructed points depends on a user by value α . If for example $\alpha = 0.01, 0.02, \dots, 0.99$, then 99 points are interpolated for each pair of nodes. The reconstructed values and interpolated points, calculated by the PNC method, are applied in the process of curve modeling. Every curve can be interpolated by some distribution function as parameter γ and nodes combination h . Parameter γ is treated as the probability distribution function for each curve.

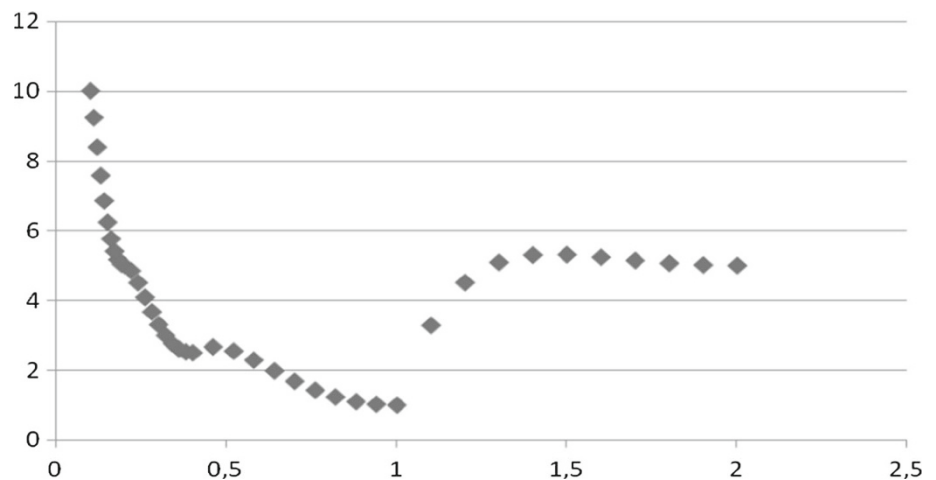
4.1 Beta distribution

Considering the probability distribution functions used nowadays for random variable $\alpha \in [0; 1]$ —one distribution deals with the range $[0; 1]$, beta distribution. The probability density function f for random variable $\alpha \in [0; 1]$ is:

$$f(\alpha) = c \cdot \alpha^s \cdot (1 - \alpha)^r, \quad s \geq 0, r \geq 0. \quad (32)$$

When $r = 0$ probability density function (32) represents $f(\alpha) = c \cdot \alpha^s$ and then probability distribution function F is like (2), for example $f(\alpha) = 3\alpha^2$ and $\gamma = \alpha^3$. If s and r

Fig. 18 Exponential distribution in handwritten symbol modeling



are positive integer numbers, then γ is the polynomial; for example $f(\alpha) = 6\alpha(1 - \alpha)$ and $\gamma = 3\alpha^2 - 2\alpha^3$. So, beta distribution gives us coefficient γ in (1) as the polynomial because of interdependence between probability density f and distribution F functions:

$$f(\alpha) = F'(\alpha), \quad F(\alpha) = \int_0^\alpha f(t) dt. \quad (33)$$

For example, (33):

$$f(\alpha) = \alpha \cdot e^{-\alpha} \quad \text{and} \quad \gamma = F(\alpha) = 1 - (1 - \alpha)e^{-\alpha}. \quad (34)$$

The basic distribution ($\gamma = \alpha$) with nodes combination $h = 0$ turns PNC interpolation (1) to linear interpolation. What about PNC in the case of yet another distribution in the range $[0; 1]$: beta distribution (32)? Power functions as γ used in Examples 1, 7 and 9–12 are also connected with beta distribution. Here are the examples of PNC modeling for beta distribution with nodes combination $h = 0$.

Example 13 PNC for $\gamma = 3\alpha^2 - 2\alpha^3$ and $h(p_1, p_2) = 0$:

Example 14 PNC for $\gamma = 4\alpha^3 - 3\alpha^4$ and $h(p_1, p_2) = 0$:

Example 15 PNC for $\gamma = 2\alpha - \alpha^2$ and $h(p_1, p_2) = 0$:

Examples 9–12 represent beta distribution with $h(p_1, p_2) = x_1y_1 + x_2y_2$.

4.2 Exponential distribution

Exponential distribution deals with random variable ≥ 0 , but in the PNC interpolation random variable $\alpha \in [0; 1]$. Then exponential distribution is represented by distribution function (34):

$$\gamma = F(\alpha) = 1 - (1 - \alpha)e^{-\alpha}.$$

Example 16 PNC for $\gamma = 1 - (1 - \alpha)e^{-\alpha}$ and $h(p_1, p_2) = 0$:

Example 17 PNC for $\gamma = 1 - (1 - \alpha)e^{-\alpha}$ and $h(p_1, p_2) = \frac{y_2}{y_1} + \frac{x_2}{x_1}$:

These examples show the variety of possibilities in curve modeling via the choice of nodes combination and probability distribution function for interpolated points.

5 Conclusions

Why and when should we use the PNC method? Interpolation methods and curve fitting represent huge problems that each individual interpolation is exceptional and requires specific solutions. The PNC method is a novel tool with all its pros and cons. The user has to decide which interpolation method is the best in a single situation. The choice is yours if you have any choice. The presented method is such a new possibility for curve fitting and interpolation when specific data (for example handwritten symbol or character) starts up with no rules for polynomial interpolation. This paper consists of two generalizations: of the previous MHR method with various nodes combinations and of linear interpolation with different (no basic) probability distribution functions and nodes combinations.

The method of probabilistic nodes combination (PNC) enables interpolation and modeling of two-dimensional curves using nodes combinations and different coefficients γ : polynomial, sinusoidal, cosinusoidal, tangent, cotangent, logarithmic, exponential, arcsin, arccos, arctan, arccot or power function, and also inverse functions. This probabilistic view is a novel approach for a problem of modeling and interpolation. Computer vision and pattern recognition are interested in appropriate methods of shape representation and curve modeling. The PNC method represents the possibilities of shape reconstruction and curve interpolation via the choice of nodes combination and probability distribution function for interpolated points. It seems to be quite new

to look at the problem of contour representation and curve modeling in artificial intelligence and computer vision.

The function for γ calculations is chosen individually at each curve modeling and treated as a probability distribution function: γ depends on the initial requirements and curve specifications. The PNC method leads to curve interpolation as handwriting modeling via a discrete set of fixed knots. So, PNC makes a combination of two important problems possible: interpolation and modeling. The main features of the PNC method are:

- a) the smaller the distance between the knots, it is better;
- b) calculations for coordinates close to zero and near extremum require more attention because of the importance of these points;
- c) PNC interpolation develops a linear interpolation into other functions as probability distribution functions;
- d) PNC is a generalization of the MHR method via different nodes combinations;
- e) interpolation of L points is connected with the computational cost of rank $O(L)$ as in the MHR method;
- f) nodes combination and coefficient γ are crucial in the process of curve probabilistic parameterization and interpolation: they are computed individually for a single curve.

Future works include application of the PNC method in signature and handwriting recognition, choice and features of nodes combinations and coefficient γ , implementation of PNC in computer vision and artificial intelligence in shape geometry, contour modelling, object recognition and curve parameterization.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

1. Lorton, A., Fouladirad, M., Grall, A.: A methodology for probabilistic model-based prognosis. *Eur. J. Oper. Res.* **225**, 443–454 (2013)
2. Pergler, M., Freeman, A.: Probabilistic modeling as an exploratory decision-making tool. McKinsey Working Papers on Risk 6, pp. 1–18 (2008)
3. Coccozza-Thivent, C., Eymard, R., Mercier, S., Roussignol, M.: Characterization of the marginal distributions of Markov processes used in dynamic reliability. *J. Appl. Math. Stoch. Anal.* 1–18 (2006). Article ID 92156
4. Jakóbczak, D.: Probabilistic modeling of signature using the method of Hurwitz–Radon matrices. *Glob. Perspect. Artif. Intell.* **1**(1), 1–7 (January 2013)
5. Collins II, G.W.: *Fundamental Numerical Methods and Data Analysis*. Case Western Reserve University, Cleveland (2003)
6. Chapra, S.C.: *Applied Numerical Methods*. McGraw-Hill, New York (2012)
7. Ralston, A., Rabinowitz, P.: *A First Course in Numerical Analysis*, 2nd edn. Dover Publications, New York (2001)
8. Zhang, D., Lu, G.: Review of shape representation and description techniques. *Pattern Recognit.* **1**(37), 1–19 (2004)
9. Ballard, D.H.: *Computer Vision*. Prentice Hall, New York (1982)
10. Tadeusiewicz, R., Flasiński, M.: *Image Recognition*. PWN, Warsaw (1991)
11. Saber, E., Yaowu, X., Murat Tekalp, A.: Partial shape recognition by sub-matrix matching for partial matching guided image labeling. *Pattern Recognit.* **38**, 1560–1573 (2005)
12. Sebastian, T.B., Klein, P.N.: On aligning curves. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(1), 116–124 (2003)
13. Liu, T., Geiger, D.: Approximate tree matching and shape similarity. In: *International Conference on Computer Vision*, Corfu-Greece (1999)
14. Choraś, R.S.: *Computer Vision*. Exit, Warsaw (2005)
15. Jakóbczak, D.: Object modeling using method of Hurwitz–Radon matrices of rank k . In: Wolski, W., Borowski, M. (eds.) *Computer Graphics: Selected Issues*, pp. 79–90. University of Szczecin Press, Szczecin (2010)
16. Jakóbczak, D.: Data extrapolation and decision making via method of Hurwitz–Radon matrices. In: Jedrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) *Computational Collective Intelligence: Technologies and Applications (Proc. of ICCCI 2011, Part 1)*. Lecture Notes in Computer Science, vol. 6922, pp. 173–182. Springer, Berlin (2011)
17. Jakóbczak, D.: Curve extrapolation and data analysis using the method of Hurwitz–Radon matrices. *Folia Oecon. Stetin. Szczec. Univ.* **9**(17), 121–138 (2011)
18. Jakóbczak, D.: Curve interpolation using Hurwitz–Radon matrices. *Pol. J. Environ. Stud.* **3B**(18), 126–130 (2009)
19. Schumaker, L.L.: *Spline Functions: Basic Theory*. Cambridge Mathematical Library, Cambridge (2007)
20. Rogers, D.F.: *An Introduction to NURBS with Historical Perspective*. Morgan Kaufmann Publishers, Menlo Park (2001)
21. Dahlquist, G., Björck, A.: *Numerical Methods*. Prentice Hall, New York (1974)
22. Dejdumrong, N.: A shape preserving verification techniques for parametric curves. *Comput. Graph. Imaging Vis.* **2007**, 163–168 (2007)
23. Dyn, N., Levin, D., Gregory, J.A.: A 4-point interpolatory subdivision scheme for curve design. *Comput. Aided. Geom. Design* **4**, 257–268 (1987)
24. Kozera, R.: *Curve Modeling via Interpolation Based on Multidimensional Reduced Data*. Silesian University of Technology Press, Gliwice (2004)
25. Jakóbczak, D.: 2D and 3D image modeling using Hurwitz–Radon matrices. *Pol. J. Environ. Stud.* **4A**(16), 104–107 (2007)
26. Jakóbczak, D.: Shape representation and shape coefficients via method of Hurwitz–Radon matrices. In: *Computer Vision and Graphics: Proceedings of ICCVG: Part I. Lecture Notes in Computer Science*, vol. 6374, pp. 411–419. Springer, Berlin (2010)
27. Jakóbczak, D.: Application of Hurwitz–Radon matrices in shape representation. In: Banaszak, Z., Świć, A. (eds.) *Applied Computer Science: Modelling of Production Processes*, vol. 1, no. 6, pp. 63–74. Lublin University of Technology Press, Lublin (2010)
28. Jakóbczak, D.: Implementation of Hurwitz–Radon matrices in shape representation. In: Choraś, R.S. (ed.) *Image Processing and Communications: Challenges 2. Advances in Intelligent and Soft Computing*, vol. 84, pp. 39–50. Springer, Berlin (2010)
29. Jakóbczak, D.: Object recognition via contour points reconstruction using Hurwitz–Radon matrices. In: Józefczyk, J., Orski, D.

- (eds.) Knowledge-Based Intelligent System Advancements: Systemic and Cybernetic Approaches, pp. 87–107. IGI Global, Hershey (2011)
30. Jakóbczak, D.: Curve parameterization and curvature via method of Hurwitz–Radon matrices. *Image Process. Commun. Int. J.* **1–2**(16), 49–56 (2011)